

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
Факультет кібербезпеки, комп'ютерної та програмної інженерії
Кафедра комп'ютерних інформаційних технологій

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач випускової кафедри
_____ А.С. Савченко
« ____ » _____ 20 ____ р

ДИПЛОМНИЙ ПРОЕКТ

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬОГО СПУПЕНЯ «БАКАЛАВР»

Тема: «Програмно-апаратні засоби керування моделлю автомобіля»

Виконавець: студент УС-412 П'ятківський Владислав Юрійович
(студент, група, прізвище, ім'я, по батькові)

Керівник: к. т. н., доцент Климова Асія Сабирівна
(науковий ступень, вчене звання, прізвище, ім'я, по батькові)

Нормоконтролер: ст. викл. Шевченко О.П.
(П.І.Б.)

(підпис)

КИЇВ 2021

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки, комп'ютерної та програмної інженерії

Кафедра комп'ютерних інформаційних технологій

Освітній ступінь: Бакалавр

Галузь знань, спеціальність, спеціалізація: 12 “Інформаційні технології”, 122 “Комп'ютерні науки”, “Інформаційні управляючі системи та технології”

ЗАТВЕРДЖУЮ

Завідувач кафедри

А.С. Савченко

“_____” _____ 2021 р.

ЗАВДАННЯ

на виконання дипломного проекту студента

П'ятківського Владислава Юрійовича
(прізвище, ім'я, по батькові)

1. Тема проекту: «Програмно-апаратні засоби керування моделлю автомобіля» затверджена наказом ректора № 636/ст. від 22.04.2021р.
2. Термін виконання роботи: з 10.05.2021 по 14.06.2021р.
3. Вихідні дані до роботи: Програмні та апаратні засоби керування моделлю автомобіля з документацією.
4. Зміст пояснювальної записки (перелік питань, що підлягають розробці): вступ, огляд технологій та методів реалізації системи, огляд компонентів системи, реалізація системи керування моделлю автомобіля.
5. Перелік обов'язкового графічного матеріалу: схематичне зображення апаратної частини системи, вигляд графічного інтерфейсу управління.

КАЛЕНДАРНИЙ ПЛАН

	Етапи виконання дипломної роботи	Термін виконання етапів	Примітка
1	Аналіз літератури та джерел за темою дипломного проекту.	11.05.2020р. – 12.05.2020р.	
2	Розробка та затвердження плану дипломного проекту.	13.05.2020р.	
3	Проведення консультації з науковим керівником щодо створення першого розділ.	14.05.2020р.	
4	Аналітичний огляд і постановка задачі.	15.05.2020р. – 18.05.2020р.	
5	Огляд технологій та методів реалізації системи.	19.05.2020р. – 22.05.2020р.	
6	Огляд компонентів системи.	23.06.2020р. – 27.05.2020р.	
7	Розробка компонентів системи.	28.05.2020р. – 04.06.2020р.	
8	Висновки та оформлення пояснювальної записки дипломного проекту.	05.06.2020р. – 08.06.2020р.	
9	Підписання необхідних документів у встановленому порядку.	09.06.2020р. – 10.06.2020р.	
10	Підготовка до захисту та попередній захист дипломного проекту на випусковій кафедрі дипломного проекту	11.06.2020р. – 14.06.2020р.	

Студент

(*П'ятківський В.Ю.*)

Керівник дипломної роботи

(*Климова А.С.*)

РЕФЕРАТ

Пояснювальна записка до дипломного проекту “Програмно-апаратні засоби керування моделлю автомобіля” містить 54 сторінки, 16 рисунків, 22 літературних джерела.

Об'єкт дослідження: програмно-апаратний засіб керування моделлю автомобіля.

Предмет дослідження: технології створення та управління роботизованими платформами.

Мета роботи: створення програмно-апаратного засобу керування моделлю автомобіля.

Методи дослідження, технічні та програмні засоби: дослідження методів та технологій розробки системи, її компонентів, створення графічного інтерфейсу для керування моделі за допомогою MIT App Inventor, створення керуючої програми для мікроконтролера Arduino UNO, тестування системи на працездатність.

Отримані результати та їх новизна: створено прототип програмно-апаратного засобу керування моделлю автомобіля з графічним інтерфейсом для керування. Матеріали дипломного проекту можуть бути використані при створенні подібних роботизованих платформ на основі мікроконтролера, оскільки кількість подібних систем постійно зростає .

ІНТЕРФЕЙС, МІКРОКОНТРОЛЕР, ДИСТАНЦІЙНЕ КЕРУВАННЯ.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ.....	7
ВСТУП.....	9
РОЗДІЛ 1. АНАЛІЗ МЕТОДІВ ТА ТЕХНОЛОГІЙ РЕАЛІЗАЦІЇ ПРОГРАМНО-АПАРАТНИХ ЗАСОБІВ КЕРУВАННЯ МОДЕЛЛЮ АВТОМОБІЛЯ.	11
1.1. Що таке мікроконтролер?	11
1.2. Різниця між мікроконтролером і мікропроцесором.....	13
1.3. Різновиди мікроконтролерів.....	15
1.4. Програмування мікроконтролерів	17
1.5. Огляд технології дистанційного керування та пультів дистанційного керування.	18
1.5.1. Використання пультів дистанційного керування.	19
1.5.2. Види пультів дистанційного керування.....	20
1.6. Висновки до розділу 1	22
РОЗДІЛ 2. АНАЛІЗ ПРОГРАМНО-АПАРАТНИХ КОМПОНЕНТІВ РЕАЛІЗАЦІЇ КЕРУВАННЯ МОДЕЛЛЮ АВТОМОБІЛЯ.	23
2.1. Огляд Arduino UNO	23
2.1.1. Живлення.....	24
2.1.2. Пам'ять.....	25
2.1.3. Порти	26
2.1.4. Зв'язок	27
2.1.5. Програмування плати Arduino.	27
2.2. Bluetooth технологія	29

2.3. Огляд окремих компонентів апаратної частини платформи.....	33
2.3.1. Bluetooth модуль для Arduino HC-05.....	33
2.3.2. Драйвер мотора.....	35
2.3.3. Сервопривід.....	36
2.4. MIT App Inventor 2.....	37
2.5. Висновки до розділу 2	39
РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНО-АПАРАТНИХ ЗАСОБІВ КЕРУВАННЯ МОДЕЛЛЮ АВТОМОБІЛЯ	40
3.1. Розробка апаратних засобів	40
3.2. Розробка програмних засобів	43
3.3. Тестування засобів керування моделлю автомобіля.....	46
Висновки до розділу 3	47
ВИСНОВКИ.....	48
СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ.....	50
ДОДАТКИ.....	52

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

Android	-	операційна система і платформа для мобільних телефонів та планшетних комп'ютерів, створена компанією Google на базі ядра Linux.
IOS	-	операційна система для iPhone, iPad та інших мобільних пристроїв Apple. Заснована на Mac OS, операційній системі, що працює на лінійці настільних та портативних комп'ютерів Mac від Apple, Apple iOS розроблена для простого та безшовного зв'язку між продуктами Apple.
UI	-	принципи та відповідний процес розробки інтерфейсу користувача для машин та програмного забезпечення при якому досягають максимально зручний спосіб їх використання користувачами.
HIGH	-	константа, рівнозначна 1, для Мікроконтролерів розпізнається як максимальна напруга, з якою вони можуть працювати.
LOW	-	константа, рівнозначна 0, для Мікроконтролерів розпізнається як мінімальна напруга, з якою вони можуть працювати.
GND	-	вузол ланцюга, потенціал якого умовно приймається за нуль, і всі напруги в системі відраховуються від потенціалу цього вузла.
VCC	-	плюс живлення щодо GND.
RXD	-	порт послідовного інтерфейсу, який використовується для отримання даних з послідовного інтерфейсу.

TXD	-	порт послідовного інтерфейсу, який використовується для надсилання даних з послідовного інтерфейсу.
Bluetooth	-	технологія бездротового зв'язку короткого радіусу дії, яка дозволяє таким пристроям, як мобільні телефони, комп'ютери та периферія, передавати дані на невеликій відстані.
AREF	-	значення опорної напруги плати.
Bootloader	-	програма, що записує скетч в пам'ять мікроконтролера.
C++	-	мова програмування високого рівня з підтримкою кількох парадигм програмування: об'єктно-орієнтованої, узагальненої та процедурної.
PWM	-	це спосіб кодування аналогового сигналу шляхом зміни ширини (тривалості) прямокутних імпульсів несучої частоти.
USB	-	(Universal Serial Bus) універсальна послідовна шина, призначена для з'єднання комп'ютерів і периферійних пристроїв.
EOM	-	(електронно-обчислювальна машина) програмно-керований пристрій для обробки інформації.
ПЗП	-	(пусково зарядний пристрій) електронний пристрій для заряду електричних акумуляторів та акумуляторних батарей енергією зовнішнього джерела.
ІЧ	-	(інфрачервоне) частина спектру випромінювання Сонця, яка безпосередньо примикає до червоної частини видимої області спектра і володіє здатністю нагрівати більшість предметів.

ВСТУП

Технології не стоять на місці і постійно розвиваються. Від створення примітивних інструментів до розробки штучного інтелекту, все було винайдено для полегшення і покращення життя.

Однією з технологій, яка уже закріпилась в нашому повсякденному житті є електронно-обчислювальні машини. Вони вже використовуються, практично у всіх сферах людською діяльності, значно полегшуючи і зменшуючи ціну на всі операції. Наряду з електронно-обчислювальними машинами стоїть така сфера, як інформаційні технології, які в тому чи іншому вигляді присутні в різних аспектах повсякденного життя людини. І ми настільки звикли до зручності цих технологій, що навіть не можемо уявити наше життя без них.

В теперішній час за допомогою такої ЕОМ пристрою як смартфон, та інформаційної технології Інтернет можна в кілька кліків можна отримати доступ практично до всієї оцифрованої бази знань людства. Та що говорити про смартфони, якщо багато інших пристроїв, таких як годинники, пілососи, холодильники тощо, отримують можливість збору та обробки інформації і мають канали для взаємодії з глобальною мережею.

Ці електронні пристрої можуть взаємодіяти між собою, створюючи канали зв'язку через мережу і обмінюючись інформацією, якою вони оперують. Відбувається збільшення рівня автоматизації середовища.

Одним з різновидів автоматизації є створення роботизованих платформ дистанційного управління. В основному, мета таких платформ є автоматизація діяльності людини, пов'язаної з ризиком для здоров'я, або навіть життя. Така система дозволить виконати важку або незручну роботу без затрат людських ресурсів. Керування відбувається за допомогою завантаженої завчасно програми, яку виконує робот, або ж за допомогою пристроїв дистанційного керування, що дозволить людині

керувати процесом виконання роботи, приймаючи задовільні рішення на кожному кроці.

Об'єктом дослідження є програмно-апаратний засіб керування моделлю автомобіля.

Предмет дослідження – технології створення та управління роботизованими платформами.

Метою даної роботи є створення програмно-апаратного засобу керування моделлю автомобіля.

Для досягнення поставленої мети потрібно:

- ознайомитись з принципами роботи дистанційного управління;
- дослідити мікроконтролер, як основний засіб реалізації моделі автомобіля;
- дослідити засоби програмування мікроконтролера;
- дослідити компоненти програмно-апаратної моделі автомобіля на дистанційному управлінні;
- реалізувати прототип платформи на дистанційному управлінні;
- побудувати графічний інтерфейс для управління моделлю автомобіля.

Практичне значення дипломного проекту полягає у наданні користувачу можливості дистанційного управління моделі автомобіля. За допомогою такої платформи також можна буде здійснювати доставку невеликих вантажів. А також це дуже непоганий варіант для дитячих розваг.

РОЗДІЛ 1

АНАЛІЗ МЕТОДІВ ТА ТЕХНОЛОГІЙ РЕАЛІЗАЦІЇ ПРОГРАМНО-АПАРАТНИХ ЗАСОБІВ КЕРУВАННЯ МОДЕЛЛЮ АВТОМОБІЛЯ.

1.1. Що таке мікроконтролер?

Мікроконтролер - це спеціальна мікросхема, призначена для управління різноманітними електричними пристроями.

Розробники мікроконтролерів придумали розумну ідею - поєднати процесор, пам'ять, ПЗП та периферію всередині одного корпусу, який виглядає як звичайна мікросхема. Відтоді виробництво мікроконтролерів у рази перевищує виробництво процесорів щороку, і попит на них не зменшується.

Мікроконтролери випускаються десятками компаній, причому випускаються не тільки сучасні 32-розрядні мікроконтролери, але і 16, і навіть 8-розрядні. У кожній родині часто можна знайти майже однакові моделі, що відрізняються швидкістю процесора та обсягом пам'яті.

Справа в тому, що мікроконтролери використовуються переважно у вбудованих системах, у іграшках, у верстатах, транспортних засобах, роботах, офісних машинах, медичних пристроях, мобільних радіоприймачах, торгових автоматах та у домашній автоматизації - там, де не потрібна потужність процесора, а баланс між ціною та достатнім функціоналом. Вони є по суті простими мініатюрними персональними комп'ютерами, призначеними для управління малими функціями більшого компонента, без складної операційної системи.

Кафедра КІТ (47)				НАУ 21 38 75 000 ПЗ			
Виконав	П'ятківський В.Ю.			Аналіз методів та технологій реалізації програмно-апаратних засобів керування моделлю автомобіля	Літера	Аркуш	Аркушів
Керівник	Климова А.С.					11	12
Консульт.					412 122		
Н-котрол.	Шевченко О.П.						
Зав. каф.	Савченко А.С.						

Ось чому найстаріші типи мікроконтролерів все ще використовуються - вони можуть зробити багато: від автоматичного відкривання дверей і включення поливу газону до інтеграції в систему розумного будинку. У той же час існують більш потужні мікроконтролери, здатні виконувати сотні мільйонів операцій в секунду і можуть бути пов'язані з безліччю периферійних пристроїв. Вони мають відповідні завдання. Таким чином, розробник спочатку оцінює завдання, а вже потім підбирає для нього відповідне обладнання.

На сьогоднішній день існує понад 200 модифікацій мікроконтролерів, сумісних з i8051, що випускаються двома десятками компаній, і велика кількість інших типів мікроконтролерів. Популярними серед розробників є 8-розрядні мікроконтролери PIC від Microchip Technology та AVR від Atmel (рис. 1.1), 16-розрядні MSP430 від TI, а також 32-розрядні мікроконтролери, архітектура ARM, яка розробляється ARM Limited та продає ліцензії іншим компаніям на їх виробництво .

Мікроконтролер характеризуються великим числом параметрів, так як він є складним пристроєм з програмним управлінням і електронним пристроєм (мікросхемою). Префікс "мікро" у назві мікроконтролера означає, що він виконується з використанням мікроелектронної технології.

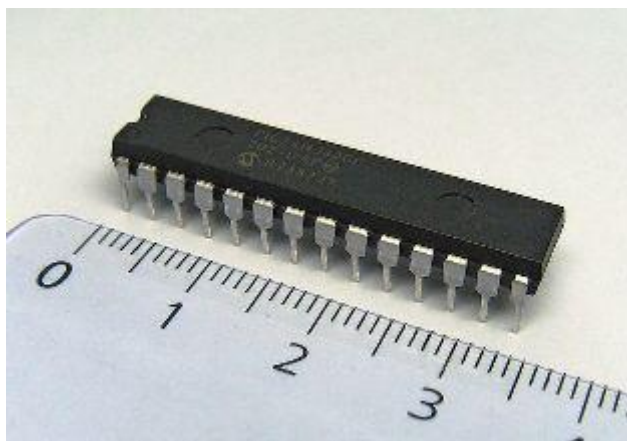


Рис.1.1. Мікроконтролер Atmel AVR ATmega8.

Під час роботи мікроконтролер зчитує команди з пам'яті або вхідного порту і виконує їх. Що означає кожна команда, визначається набором команд

мікроконтролера. Система інструкцій вбудована в архітектуру мікроконтролера, а виконання командного коду виражається у здійсненні певних мікрооперацій внутрішніми елементами мікросхеми.

Мікроконтролери дозволяють гнучко керувати різними електронними та електричними пристроями. Деякі моделі мікроконтролерів настільки потужні, що можуть безпосередньо перемикає реле (наприклад, на гірляндах ялинки).

Зазвичай пам'ять у мікроконтролерах коливається від 2 до 128 КБ. Перш ніж нарешті програмувати мікроконтролер, він тестується в емуляторах - програмному чи апаратному.

1.2. Різниця між мікроконтролером і мікропроцесором.

Тут може виникнути запитання: чи є мікропроцесор та мікроконтролер просто різними назвами одного і того ж пристрою, чи це все одно різні речі?

Мікропроцесор - це центральний пристрій будь-якого комп'ютера, виготовлений за інтегральною технологією. Сама назва говорить про те, що саме в ній відбуваються обчислювальні процеси. Щоб зробити з нього комп'ютер, навіть не дуже сучасний і потужний, його потрібно доповнити зовнішніми пристроями. Перш за все, це оперативна пам'ять та порти для введення / виведення інформації.

Мікроконтролер має в собі процесор, оперативну пам'ять, програмну пам'ять і, крім того, цілий набір периферійних пристроїв, які перетворюють процесор на повністю функціональний комп'ютер. За старою термінологією радянських часів такі пристрої називали одночиповими мікро ЕОМ. Але, як відомо, радянські комп'ютерні технології зайшли в глухий кут, а разом з ним і ОМЕОМ.

Іноземні комп'ютерні технології не стояли на місці, тому ОМЕОМ стали називати контролерами (від англійського Control - керувати, контролювати). Дійсно, контролери виявилися цілком придатними для управління різним обладнанням, навіть не дуже складним.

Центральний процесор у кожному комп'ютері - це головний комп'ютер. Хоча комп'ютер не призначений виключно для обчислювальної роботи, в основі його лежить процесор. Але це не просто комп'ютер, який має процесор.

Якщо задуматися і придивитися, то можна виявити, що процесори використовуються в більшості побутових приладів. Тільки там використовуються не такі процесори, як у комп'ютері, а мікропроцесори і навіть мікроконтролери.

Отже, що таке мікроконтролер і чим він відрізняється від самого мікропроцесора, чи це абсолютно різні електронні компоненти?

Великі інтегральні схеми або високоінтегровані мікросхеми є процесорами. Мікропроцесори - це, по суті, ті самі процесори, але через префікс "мікро" визначається їх суть, що вони менші за своїх "великих" аналогів. За свого історичного часу процесор із своїми розмірами міг займати більше однієї кімнати.

Зменшений розмір і зібраний процесор займає менше місця, і його можна помістити в більш компактний виріб, це мікропроцесор. Але сам процесор не здатний робити багато іншого, крім передачі даних між регістрами та виконання деяких арифметичних та логічних операцій над ними.

Для того, щоб мікропроцесор передавав дані в пам'ять, саме ця пам'ять повинна бути присутнім або на самому кристалі, на якому розташований сам процесорний елемент, або бути підключеною до зовнішньої оперативної пам'яті, виконаної у вигляді окремого кристала або модуля.

Крім пам'яті, процесор повинен взаємодіяти із зовнішніми пристроями - периферійними пристроями. В іншому випадку, чого хорошого можна очікувати від процесора, перемішування та переміщення даних вперед-назад. Сенс виникає, коли процесор взаємодіє з пристроями вводу-виводу. У комп'ютері це клавіатура, маніпулятор миші та пристрої відображення як дисплей, за бажанням принтер і, наприклад, сканер, знову для введення інформації.

Для управління пристроями вводу-виводу необхідні відповідні буферні схеми та елементи. На їх основі реалізовано так зване апаратне забезпечення інтерфейсу. Методи взаємодії з елементами інтерфейсу передбачають наявність схем портів

вводу-виводу, декодерів адрес та драйверів шини з буферними схемами для збільшення навантажувальної здатності мікропроцесора.

Інтеграція процесора з усіма необхідними додатковими елементами для того, щоб цей продукт був залитий в якусь цілісну конструкцію і призводить до утворення мікроконтролера. Мікросхема або мікроконтролерний чіп реалізує процесор та інтерфейсні схеми на одній матриці.

Самодостатній чіп, який містить майже все, що достатньо для побудови готового виробу, і є прикладом типового мікроконтролера. Наприклад, електронні наручні годинники або будильник мають всередині мікроконтролер, який реалізує всі функції такого пристрою. Окремі периферійні пристрої підключені безпосередньо до ніжок мікросхеми мікроконтролера, або разом використовуються додаткові елементи або мікросхеми малого або середнього ступеня інтеграції.

Мікроконтролери широко використовуються у виробках, що містять цілу систему виключно в одній мініатюрній мікросхемі, яку часто називають мікробіркою. Наприклад, кредитна картка "чіп" містить мікроконтролер всередині пластикової основи. Планшет домофона також містить мікроконтролер всередині. А приклади використання та застосування мікроконтролерів настільки великі в сучасному світі, що легко виявити наявність контролера в будь-якому більш-менш інтелектуальному пристрої - від дитячої іграшки до бездротової гарнітури для стільникового телефону.

1.3. Різновиди мікроконтролерів.

Мікроконтролери можна розділити за такими критеріями:

- розрядність;
- система команд;
- архітектура пам'яті.

Розрядність - це довжина одного слова оброблюваного контролером або процесором. Розподіл відбувається на: 8-біт, 16-біт, 32-біта, 64-біта. Чим розрядність

більша, тим швидше мікроконтролер може обробити великі масиви даних, але такий підхід не завжди справедливий, Для кожного завдання висувуються індивідуальні вимоги, як за швидкістю, так і за способом обробки, наприклад, застосування 32-х розрядного ARM мікропроцесора для роботи в простих пристроях, що оперують з 8 бітними словами може бути не обґрунтоване як по зручності написання програми і обробки інформації, так і по собі вартості.

Однак, за статистикою на 2017 рік, вартість таких контролерів активно знижується, і, якщо так триватиме і далі - він буде дешевше найпростіших RISC контролерів, при наявності набагато більшому наборі функцій. Не зрозуміло тільки одне - це маркетинговий хід і зниження ціни, або реальний технологічний прогрес.

Розподіл за типом системи команд:

- RISC-архітектура, або скорочена система команд. Орієнтована на швидке виконання базових команд за 1, рідше 2 машинних цикли, а також має велику кількість універсальних регістрів, і довший спосіб доступу до постійної пам'яті. Архітектурна характерна для систем під управління UNIX;
- CISC-архітектура, або повна система команд, характерна пряма робота з пам'яттю, більше число команд, мале число регістрів (орієнтована на роботу з пам'яттю), тривалість команд від 1 до 4 машинних циклів. Приклад - процесори Intel.
- Розподіл за типом пам'яті:
- Архітектура Фон-Неймана - має загальну область пам'яті для команд і даних, при роботі з такою архітектурою в результаті помилки програміста дані можуть записатися в область пам'яті програм і подальше виконання програми стане неможливим. Пересилання даних і вибірка команди не може здійснюватися одночасно з тих же причин. Розроблено в 1945 році.
- Гарвардська архітектура - роздільна пам'ять даних і пам'ять програм, використовувалася в перші на комп'ютерах сімейства Mark. Розроблено в 1944 році.

1.4. Програмування мікроконтролерів

Мікроконтролери спочатку програмувались лише на мові асемблера, але різні мови програмування високого рівня, такі як C, Python та JavaScript, зараз також широко використовуються для програмування мікроконтролерів та вбудованих систем. Компілятори для мов загального призначення, як правило, матимуть деякі обмеження, а також вдосконалення для кращої підтримки унікальних характеристик мікроконтролерів. Деякі мікроконтролери мають середовища, які допомагають розробляти певні типи програм (рис. 1.2). Постачальники мікроконтролерів часто роблять інструменти у вільному доступі, щоб полегшити використання обладнання.

Симулятори доступні для деяких мікроконтролерів. Вони дозволяють розробнику проаналізувати, якою має бути поведінка мікроконтролера та їх програми, якщо він використовував фактичну деталь. Симулятор покаже внутрішній стан процесора, а також стан виходів, а також дозволить генерувати вхідні сигнали. Хоча, з одного боку, більшість тренажерів не зможуть імітувати багато іншого обладнання в системі, вони можуть виконувати умови, які в іншому випадку важко відтворити за власним бажанням у фізичній реалізації, і можуть бути найшвидшим способом налагодження та аналізу проблеми.

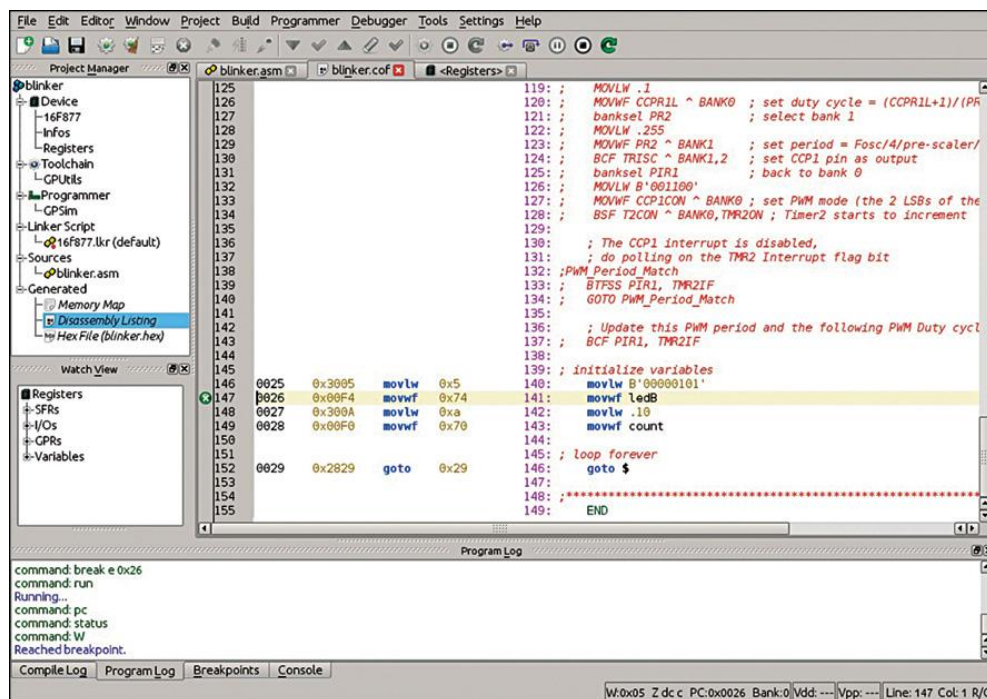


Рис. 1.2. Piklab – інтегроване середовище розробки для мікроконтролерів сімейства Pīc.

1.5. Огляд технології дистанційного керування та пультів дистанційного керування.

Одним з найважливіших компонентів сучасних ЕОМ є пристрій вводу-виводу. Це пристрій для обміну інформацією між комп'ютером та зовнішнім середовищем. Якщо в перших обчислювачах введення інформації здійснювався за допомогою прямого замикання блоків пам'яті, то згодом, в більш пізніших машинах, з'явився такий пристрій введення як клавіатура. Пізніше ще з'явилися такі пристрої як миша та графічний планшет.

В даний час все більше з'являється пристроїв, що працюють дистанційно - так званих «розумних» пристроїв. Однією з головних функцій такого пристрою є дистанційне керування (в найпростішому випадку включення і відключення) будь-якого обладнання. З'являються спеціальні програми за допомогою, яких людина може управляти електроустаткуванням. За допомогою смартфона з використанням WI-FI

мережі можна управляти як побутовими приладами, так і великими об'єктами виробничих комплексів.

Дистанційне керування - це передача сигналу від оператора до об'єкта управління, який розташований на певній відстані. Найчастіше використовується, коли доступ до об'єкта управління неможливий з яких-небудь причин або небажаний, а також коли об'єкт знаходиться в русі. Пульти дистанційного керування – це електричний пристрій, який використовується для керування іншим пристроєм на певній відстані.

1.5.1. Використання пультів дистанційного керування.

Звичайно, пульти дистанційного керування - це не просто серфінг по каналах. У наші дні ви можете знайти можливість дистанційного керування, вбудовану у величезний набір продуктів. Іграшкові автомобілі та вертольоти, приставки для відеоігор, стельові вентилятори, як вам сказати - є велика ймовірність, що ви знайдете версію, яка управляється за допомогою пульта дистанційного керування. Ви навіть можете придбати унітаз з дистанційним управлінням.

А самі пульти дистанційного керування бувають найрізноманітніших форм і розмірів, починаючи від великогабаритних версій для короткозорих і закінчуючи крихітними пристроями кишенькового розміру. Світлодіодний телевізор Samsung 9000 навіть оснащений пультом дистанційного керування, який має вбудований сенсорний РК-екран, що дозволяє переглядати телевізійну програму, відмінну від програми на великому екрані.

Смартфони стають універсальними пультами дистанційного керування, якими можна керувати безліччю цифрових продуктів. За допомогою потрібного додатка ви можете за допомогою телефону розблокувати двері автомобіля на милі, запланувати свій відеореєстратор для запису телевізійної програми, керувати YouTube на своєму ноутбукі або, як ви вже здогадалися, змінити канали (і багато іншого) на своєму телевізорі.

Віддалені технології теж мають більш серйозні цілі. Зараз у всьому світі використовуються різні боєприпаси з точним наведенням, що використовуються в конфліктах. Лазерно-керовані бомби застосовуються для ураження невеликих ділянок, на які було набагато важче атакувати, використовуючи звичайні "німі" бомби. Крилаті ракети можна запускати на відстані багатьох миль і направляти майже в будь-яку ціль.

Зараз різноманітна броньована техніка оснащена баштами з дистанційним управлінням, які дозволяють солдатам сидіти під зброєю у відносній безпеці, націлюючи та стріляючи з пістолета за допомогою камери та джойстика. Безпілотні літальні апарати забезпечують дистанційне спостереження та можливості атаки, керовані ударними командами, пов'язаними з офісами, за тисячі миль від них.

Дистанційна технологія дозволяє нам переслідувати менш руйнівні прагнення. У багатьох своїх проектах NASA значною мірою покладається на дистанційне управління. Один з найбільших тріумфів організації припав на 1997 рік, коли космічний корабель Mars Pathfinder розмістив на поверхні Марса рухомий зонд.

Вчені на Землі надіслали інструкцію до марсохода, наказавши йому використовувати різні інструменти для збору даних про погоду, ґрунтові умови та багато іншого. Успіх цієї місії породив подальшу місію в 2003 році, коли ровери Spirit і Opportunity роками досліджували Марс.

1.5.2. Види пультів дистанційного керування.

Сьогодні на ринку існує безліч типів пультів дистанційного керування, але чи замислювались ви коли-небудь про те, як працюють ці пульти дистанційного керування, або яка технологія використовується.

Інфрачервоні або ІЧ-пульти використовують невидиме інфрачервоне світло. Ці інфрачервоні лампочки модулюються і встановлюються на певну частоту для надсилання унікальних спалахів ІЧ світла на пристрій. Потім кожен інфрачервоний код інтерпретується як певна функція ІЧ-приймачем керованого пристрою і

представляє різні команди, такі як увімкнення / вимкнення живлення, збільшення / зменшення гучності або збільшення / зменшення каналу та багато іншого. Більшість електронних пристроїв, якими керують пульти дистанційного керування, використовують інфрачервону технологію. Це характерно для телевізорів, супутникових приймачів, кабельних коробок, відео-регістраторів, пристроїв домашньої автоматизації тощо. Інфрачервона технологія також використовується для управління механічними пристроями, такими як підйомники та двигуни.

Переваги: проста схема керування, низька вартість, більший час автономної роботи.

Недоліки: пульт потрібно спрямовувати на керуючий пристрій.

Радіочастотний пульт дистанційного керування - це портативний бездротовий пристрій, який використовується для управління різними електронними пристроями (аудіо, відео та іншим електронним обладнанням) за допомогою радіочастотної передачі. Радіочастотні пульти не потрібно спрямовувати на потрібний пристрій, як це відбувається в ІЧ-пультах.

Радіочастотна технологія також широко використовується в наші дні, проте вона все ще рідша порівняно з ІЧ технологією керування. Ця технологія працює на модульованій хвильовій формі в радіочастотному спектрі. Радіочастотний пульт дистанційного керування передає унікальні коди на головний пристрій. Приймаючий пристрій складається з радіочастотного приймача, який здатний розпізнавати радіочастотний сигнал від радіочастотного пульта, а потім декодує ці сигнали для виконання певної дії.

Радіочастотні пульти дистанційного керування дуже часто використовуються для відкривання гаражних дверей, автосигналізації, радіокерованих іграшок. Вони також використовуються в домашніх розважальних пристроях, супутникових приймачах, телевізорах, високоякісних стереосистемах та інших дистанційно керованих пристроях.

Переваги: широкий діапазон дії, немає перешкод.

Пульт дистанційного керування Wi-Fi передає сигнали управління через мережу Wi-Fi. Ці пульти дистанційного керування використовують технологію радіочастот. Виробництво пульта дистанційного керування Wi-Fi є більш складним, ніж інші стандартні пульти дистанційного керування. Ці пульти слід спочатку підключити до мережі Wi-Fi, оскільки вони використовують протокол Wi-Fi для роботи.

Переваги: можна підключити навіть смартфон.

Недоліки: керований пристрій і пульт мають знаходитись в одній мережі.

Дротові пульти дистанційного керування фізично підключені до пристрою для управління. Провідний пульт дистанційного керування надсилає команду на пристрій через підключений провід.

Переваги: пульт напряму передає команди в керований пристрій без перешкод.

Недоліки: можуть працювати лише з пристроєм, підключеним через провід.

1.6. Висновки до розділу 1

Проаналізувавши викладений в розділі вище матеріал, можна сказати, що для реалізації невеликих комп'ютеризованих платформ не обов'язково використовувати звичайні плати і процесори, які ми звикли бачити в ПК. Для таких проектів краще підходить мікроконтролер, який є дуже зменшеною версією комп'ютера. А способи керування таких платформ є різноманітними.

РОЗДІЛ 2

АНАЛІЗ ПРОГРАМНО-АПАРАТНИХ КОМПОНЕНТІВ РЕАЛІЗАЦІЇ КЕРУВАННЯ МОДЕЛЛЮ АВТОМОБІЛЯ.

2.1. Огляд Arduino UNO

Arduino / Genuino Uno - це плата мікроконтролера на базі процесора ATmega328P. Максимальна довжина і ширина друкованої плати Uno становить 6.9 см і 5.4 см відповідно, з урахуванням роз'єму USB і роз'єму живлення, які виступають за межі плати. Вона має 14 цифрових входів / виходів (з яких 6 можна використовувати як ШІМ-виходи), 6 аналогових входів, чіп з частотою роботи 16 МГц, два роз'єми: силовий і USB, роз'єм ISCP для внутрисхемного програмування і кнопку перезавантаження пристрою (рис. 2.1). Для стабільної роботи плату необхідно підключити до живлення або через вбудований USB Роз'єм, або підключивши роз'єм живлення до джерела від 7 до 12В. Через перехідник живлення плата також може працювати і від батареї формату Крона.

Плата Arduino / Genuino Uno може живитися через USB-з'єднання або за допомогою зовнішнього джерела живлення. Джерело живлення вибирається автоматично. Зовнішнє (не USB) живлення може надходити або від адаптера змінного струму до постійного струму, або від акумулятора. Адаптер можна підключити, підключивши 2,1-мм позитивний штекер до центрального гнізда в гнізді живлення плати. Виводи від акумулятора можна вставити в роз'єми штепсельних роз'ємів GND та Vin.

Кафедра КІТ (47)				НАУ 21 38 75 000 ПЗ			
Виконав	П'ятківський В.Ю.			Аналіз програмно-апаратних компонентів реалізації керування моделлю автомобіля.	Літера	Аркуш	Аркушів
Керівник	Климова А.С.					23	17
Консульт.					412 122		
Н-котрол.	Шевченко О.П.						
Зав. каф.	Савченко А.С.						

Плата може працювати від зовнішнього джерела живлення від 6 до 20 вольт. Однак якщо джерело живлення менше 7 В, штифт 5 В може подавати напругу менше п'яти вольт, і плата може стати нестійкою. Рекомендується подавати напругу в діапазоні від 7В до 20В, для уникнення перегріву і згоряння вбудованого стабілізатора.

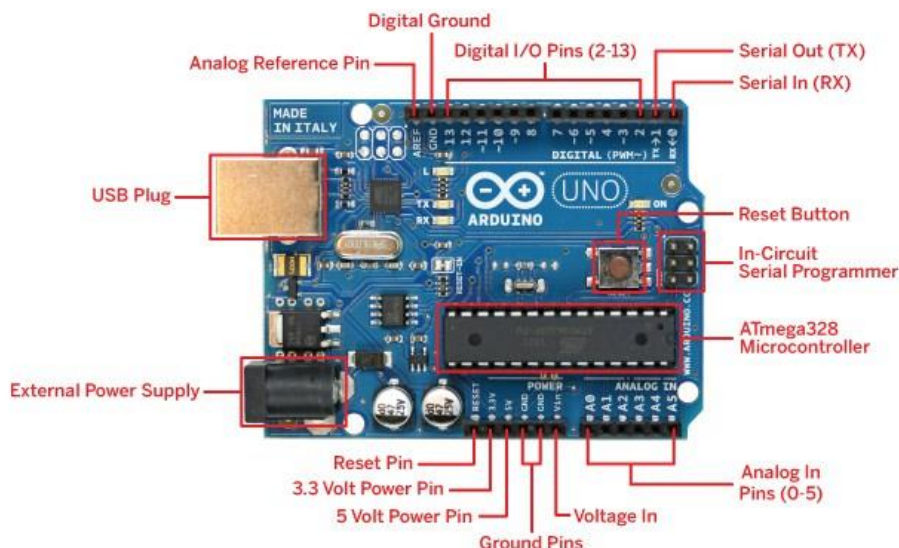


Рис. 2.1. Плата Arduino UNO R3.

2.1.1. Живлення

На платі передбачено кілька портів, що дозволяють жити від неї підключені датчики, сенсори і актуатори. Всі ці порти позначені:

- Vin. Вхід живлення. Використовується для отримання живлення від зовнішнього джерела. Через дайний порт відбувається тільки подача живлення на плату, отримати звідти живлення для зовнішніх пристроїв неможливо
- 5V. Джерело 5В напруги для живлення зовнішніх пристроїв. При отриманні живлення платою з будь-яких інших джерел (USB, роз'єм живлення або Vin) на цьому контакті ви завжди зможете отримати

стабільну напругу 5 вольт. Його можна вивести на макетну плату або подати безпосередньо на свій пристрій.

- 3V3. Джерело 3.3 В напруги для живлення зовнішніх пристроїв. Працює за таким-же принципом, що і контакт 5V. З даної ніжки також можна вивести напруга на макетну плату, або подати на необхідний датчик / сенсор напруги. Максимальний обсяг струму становить 50 мА.
- GND. Контакт для підключення заземлення. Необхідний для створення замкненого кола при підключенні до контактів Vin, 5V або 3V3. У всіх випадках ніжку GND необхідно виводити як мінус, інакше ланцюг не буде замкнутий і живлення не буде подаватися.
- IOREF. Цей порт на платі Arduino / Genuino забезпечує еталон напруги, з яким працює мікроконтролер. Залежно від напруги на порту IOREF, плата може переключитися на відповідне джерело живлення або задіяти перетворювачі рівнів, що дозволить їй працювати як з 5В, так і з 3.3В-пристроями.

Для захисту USB порту комп'ютера від зворотних струмів, короткого замикання і перенавантаження, на платформі Arduino Uno вбудований автоматичний самовостанавлюючийся запобіжник. При проходженні струму харчування більше 500 мА через USB порт, запобіжник автоматично спрацьовує і розмикає ланцюг живлення до тих пір, поки значення струму не повернутися до нормальних.

2.1.2. Пам'ять

Плата Arduino Uno за замовчуванням підтримує три типи пам'яті:

- Flash - пам'ять об'ємом 32 кБ. Це основне сховище для команд. Коли ви прошиває контролер своїм скетчем, він записується саме сюди. 0.5кб з даного пулу пам'яті відводиться на bootloader - програму, яка займається ініціалізацією системи, завантаження через USB і запуску скетчу.

- Оперативна SRAM пам'ять об'ємом 2 кБ. Тут по-замовчуванню зберігаються змінні і об'єкти, створювані в ході роботи програми. Пам'ять ця енерго-залежна, при виключенні живлення всі дані, зрозуміло, зітруться.
- Незалежна пам'ять (EEPROM) обсягом 1кб. Тут можна зберігати дані, що не зітруться при виключенні контролера. Але процедура запису і зчитування EEPROM вимагає використання додаткової бібліотеки, яка доступна в Arduino IDE за замовчуванням. Також ніжно пам'ятати про обмеження циклів перезапису, властивих технології EEPROM.

2.1.3. Порти

Порти з номерами від 0 до 13 є цифровими. Це означає, що ви можете зчитувати і подавати на них тільки два види сигналів: HIGH (логічна 1) і LOW (логічний 0). HIGH сигнал розпізнається Arduino як струм напругою 5В, а LOW сигнал, відповідно 0В. Кожен з 14 цифрових контактів на Uno може використовуватися як вхід або вихід. Кожен порт може забезпечувати або приймати 20 мА, як рекомендовано, і має внутрішній підтягуючий резистор (відключений за замовчуванням) 20-50 кОм. Максимум 40 мА - це значення, яке не можна перевищувати на жодному контакті вводу-виводу, щоб уникнути постійних пошкоджень мікроконтролера.

Arduino Uno має на своїй платформі 6 аналогових входів з дозволом 10 Біт на кожен вхід. Даний дозвіл говорить нам про те, що сигнал, що приходить на нього, можна оцифрувати в діапазоні від 0 до 1024 умовних значень.

Окрім цього деякі порти можуть виконувати додаткові функції. 0(RX) і 1(TX) цифрові порти послідовного інтерфейсу використовуються для надсилання та отримання інформації через послідовний інтерфейс (засіб зв'язку з іншими пристроями). Порти 2 та 3 можуть використовуватися для виклику переривань в роботі мікроконтролера при низькому рівні сигналу на цих контактах. Цифрові контакти 3, 5, 6, 9, 10, 11 можуть виводити 8-бітні аналогові значення у вигляді ШІМ-

сигналу. На порту 13 є вбудований світлодіод, який включається при отриманні HIGH і відключається при отриманні LOW сигналу.

2.1.4. Зв'язок

Arduino Uno надає ряд можливостей для здійснення зв'язку з комп'ютером, ще одним Arduino або іншими мікроконтроллерами. У ATmega328 є приймач UART, що дозволяє здійснювати послідовний зв'язок за допомогою цифрових портів 0 (RX) і 1 (TX). Мікроконтролер ATmega16U2 на платі забезпечує зв'язок цього приймача з USB-портом комп'ютера, і при підключенні до ПК дозволяє Ардуіно визначатися як віртуальний COM-порт. Прошивка мікросхеми 16U2 використовує стандартні драйвера USB-COM, тому установка зовнішніх драйверів не потрібна. У пакет програмного забезпечення Ардуіно входить спеціальна програма, що дозволяє зчитувати і відправляти на Ардуіно прості текстові дані. При передачі даних через мікросхему-перетворювач USB-UART під час USB-з'єднання з комп'ютером, на платі будуть мигати світлодіоди RX і TX. Також є вбудована бібліотека SoftwareSerial дозволяє реалізувати послідовний зв'язок на будь-яких цифрових висновках Arduino Uno, якщо вбудовані порти 0 та 1 будуть використовуватись для інших потреб.

2.1.5. Програмування плати Arduino.

Для написання програм (скетчів) для контролер Ардуіно вам потрібно встановити середовище програмування. Найпростішим варіантом буде установка безкоштовної Arduino IDE, скачати її можна з офіційного сайту.

Сам скетч найчастіше представляє собою нескінченний цикл, в якому регулярно перевіряються порти з приєднаними до них датчиками і за допомогою спеціальних команд формується керуючий вплив на зовнішні пристрої (вони включаються або вимикаються). У програміста Ардуіно є можливість підключити готові бібліотеки, як вбудовані в IDE, так і доступні на численних сайтах і форумах.

Написана і скомпільована програма завантажується через USB-з'єднання (UART- Serial). З боку контролера за цей процес відповідає bootloader.

Ардуіно програмується на мові програмування C / C ++ з відповідним йому синтаксисом. Вбудований складальник, препроцесор і компілятор виправляють велику кількість помилок і робить багато за користувача автоматично, ми навіть про це не знаємо і не замислюємося. Базові функції для управління портами і інтерфейсами мікроконтролера, математика і деякі інші функції / макроси взяті з відкритого фреймворка для роботи з мікроконтролерами під назвою Wiring. Саме з нього складається базовий набір інструментів Ардуіно. У зв'язку з цим самі розробники Ардуіно називають мову "спрощеним c ++", і навіть дали йому окрему назву - Arduino Wiring. З самого початку в Arduino IDE нам доступна величезна купа різних функцій і інструментів: Всі можливості мови C ++, які надає компілятор: типи даних, оператори і взагалі весь неосяжний синтаксис. Ми програмуємо на тому ж C ++, на якому можна програмувати в будь-якому іншому місці. "Ядро" Ардуіно - бібліотека Arduino.h, яка автоматично підключається в код. У ній містяться функції для управління пінами, інтерфейсами, а також є набір всяких корисних функцій і інструментів. А ще воно відповідає за ініціалізацію і настройку периферії мікроконтролера при запуску. В ядрі, до речі, лежать стандартні бібліотеки для Serial, Wire, SPI і EEPROM. В папці з програмою лежить набір стандартних бібліотек: для LCD дисплея, крокового мотора, сервоприводу і деяких інших приладів. З компілятором йде набір низькорівневих бібліотек для AVR. Компілятор дозволяє працювати з мікроконтролером "безпосередньо" за допомогою регістрів і читання даташіта. Також ми можемо писати на асемблері, взявши під контроль кожен такт роботи мікроконтролера.

При запуску Arduino IDE дає нам заготовку у вигляді двох обов'язкових функцій: setup() і loop(). Код в блоці setup() виконується один раз при кожному запуску мікроконтролера. Код в блоці loop() виконується "по колу" на всьому протязі роботи мікроконтролера, починаючи з моменту завершення виконання setup(). Також програма може містити підключення бібліотек. Бібліотека є файлом (набором

файлів), що містить такий самий C ++ код, на якому ми пишемо скетч (іноді зустрічаються і асемблерні вставки). Ми можемо підключити бібліотеку в свій код і використовувати можливості, які вона дає, а варіантів там дуже багато: готові "інструменти" для роботи з зовнішніми датчиками і модулями, для роботи з внутрішньою периферією мікроконтролера (таймери, АЦП, пам'ять), бібліотеки різних математичних інструментів і багато чого іншого.

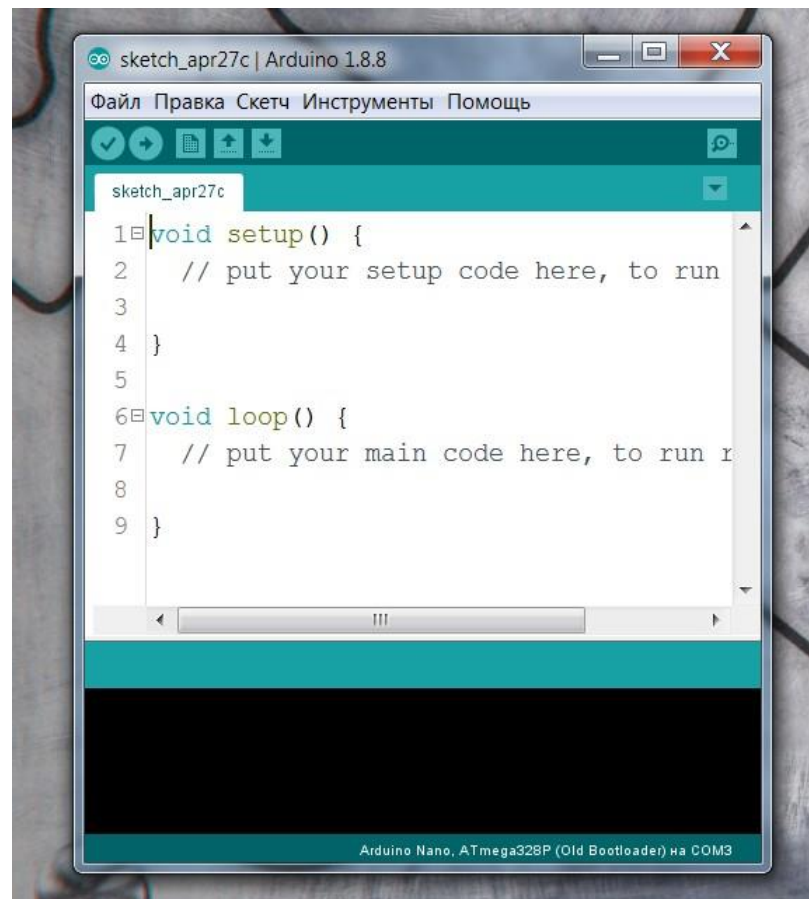


Рис. 2.2. Інтерфейс середовища Arduino IDE.

2.2. Bluetooth технологія

Bluetooth - це технологія бездротового з'єднання з іншими пристроями на відстані до 100 метрів. Роботи зі створення Bluetooth були розпочаті в 1994 році виробником телекомунікаційного устаткування Ericsson в якості бездротової альтернативи кабелям RS-232. В результаті Bluetooth розробила група Bluetooth

Special Interest Group, заснована в 1998 році. До неї увійшли компанії Ericsson, IBM, Intel, Toshiba і Nokia. Згодом Bluetooth стала частиною міжнародного стандарту IEEE 802.15.1. Саме слово Bluetooth - це переклад на англійську мову датського слова "Blatand" ("Синьозубий"). Так прозвали короля вікінгів Харальда I, який об'єднав ворогуючі датські племена в єдине королівство. Bluetooth по суті робить те ж саме з протоколами зв'язку, об'єднуючи їх в один універсальний стандарт.

Інтерфейс Bluetooth дозволяє передавати як голос (зі швидкістю 64 Кбіт / сек), так і дані. Для передачі даних можуть бути використані асиметричний (721 Кбіт / сек в одному напрямку і 57,6 Кбіт / сек в іншому) і симетричний методи (432,6 Кбіт / сек в обох напрямках). Bluetooth чіп - це маленький високочастотний (2.4 - 2.48 ГГц) приймач (рис. 2.3), який дозволяє в залежності від ступеня потужності встановлювати зв'язок межах 10 або 100 метрів. Різниця в відстані, безумовно, велика, однак з'єднання в межах 10 м дозволяє зберегти низьке енергоспоживання, компактний розмір і досить невисоку вартість компонентів. Так, малопотужний передавач споживає всього 0.3 мА в режимі standby і в середньому 30 мА при обміні інформацією. Як радіотехнологія, Bluetooth здатний «обходити» перешкоди, тому що з'єднуються, можуть знаходитися поза зоною прямої видимості. З'єднання відбувається автоматично, як тільки Bluetooth-пристрої виявляються в межах досяжності, причому не тільки за принципом точка – точка (два пристрої), але і за принципом точка – багато точок (один пристрій працює з декількома іншими).

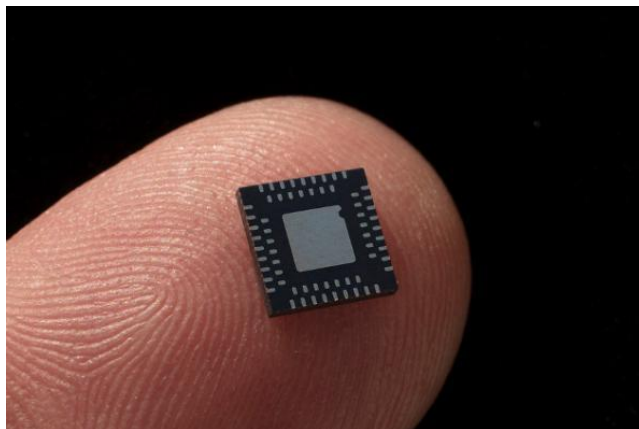


Рис. 2.3. Bluetooth чіп.

Принцип дії Bluetooth заснований на використанні радіохвиль. При включенні Bluetooth активується радіопередавач, який працює в обмеженому діапазоні частот в районі 2,4 ГГц. Ця частина спектра називається ISM - Industry, Science and Medicine - і використовується в різних побутових приладах і бездротових мережах. Після активації радіопередавач починає відстежувати всі сигнали в цьому діапазоні. Другий пристрій робить те ж саме. Після того, як пристрої виявили один одного, перше бере на себе роль передавача, а друге стає приймачем. При цьому дані передаються за спеціальним алгоритмом FHSS, який забезпечує стійкість до широкосмугових перешкод. Згідно з цим алгоритмом, частота сигналу Bluetooth стрибкоподібно змінюється 1600 разів в секунду, перестрибуючи з однієї з наявних 79 частот на іншу. Послідовність перемикання між частотами для кожного з'єднання є псевдослучайною і відома тільки передавача і приймача, які кожні 625 мікросекунд синхронно перебудовуються з однієї частоти на іншу. Таким чином, якщо поруч працюють кілька пар приймачів-передавачів, то вони не заважають один одному. Цей алгоритм також є частиною системи захисту інформації, що передається.

Перед відправкою по Bluetooth дані розбиваються на спеціальні блоки, іменовані пакетами, з інструкціями по складанню з них вихідного файлу. Процесор приймача обробляє пакети, створює з них передається файл і поміщає його в постійну пам'ять пристрою.

При передачі цифрових даних і аудіосигналу використовуються різні схеми кодування: аудіосигнал не повторюється, а цифрові дані в разі втрати пакета інформації передаються повторно.

Коли пара будь-яких Bluetooth-пристроїв з'єднується, то вони утворюють пікомережу. Одне з них, яке виконує функції ведучого пристрою, формує сигнали синхронізації частоти і її зміни. Зазвичай провідним є той модуль, який розміщений в найбільш потужному пристрої, такому, як персональний комп'ютер. Всі інші пристрої є відомими. Пікомережа є фундаментальною формою комунікації в технології Bluetooth. Пікомережа може містити до 7 активних ведучих пристроїв.

Крім того, в зоні впевненого прийому провідного пристрою можуть перебувати неактивні ведені пристрої, які також синхронізовані на загальні годинник і загальну послідовність зміни частот, але не можуть обмінюватися даними до тих пір, поки ведучий пристрій не активує їх. Якщо в мережі виявляється більше 8 пристроїв, то буде сформована друга пікомережа і так далі. Кілька (до 10) незалежних і навіть не синхронізованих між собою пікомереж, між якими можливий обмін інформацією, можуть об'єднуватися в так звану велику мережу Scatternet. Для цього кожна пара пікомереж повинна мати як мінімум одну спільну пристрій, який буде головним в одній і підпорядковується в іншій. Таким чином, в межах окремої Scatternet може бути одночасно пов'язано максимум 71 пристрій.

Говорячи про бездротовий зв'язок, не можна залишити поза увагою питання безпеки такого з'єднання. Крім фокуса з частотними шаблонами і необхідності синхронізації прийому-передачі в стандарті Bluetooth передбачено шифрування даних, що передаються з ключем ефективної довжини від 8 до 128 біт і можливістю вибору односторонньої або двосторонньої аутентифікації (звичайно, можна обійтися взагалі без аутентифікації), що дозволяє встановлювати стійкість результуючого шифрування відповідно до законодавства кожної окремої країни (в деяких країнах заборонено використання сильної криптографії). На додаток до шифрування на рівні протоколу може бути застосоване шифрування на рівні додатків.

Часто доводиться стикатися з думкою, що Bluetooth-пристрої, які знаходяться в межах дії зв'язку, можуть просто з'єднатися і почати обмінюватися конфіденційною інформацією. Насправді автоматичний обмін інформацією між Bluetooth-пристроями ведеться лише на рівні апаратного забезпечення, тобто виключно для визначення самого факту можливості з'єднання. А ось на рівні додатків користувач сам вирішує, ввести або заборонити автоматичну установку зв'язку. Таким чином, використання Bluetooth стає безпечніше підключення до Інтернету, в якому всі вузли також з'єднані електрично, але це ще не означає отримання беззастережного доступу до будь-якого ресурсу.

Варто також зауважити, що стандарт Bluetooth розроблявся з розрахунком на малу потужність, тому вплив його на організм людини зведено до мінімуму.

Bluetooth пристосований для використання в тих бездротових пристроях зв'язку, де потрібно досить низька ціна, немає необхідності у високих швидкостях і бажано низьке енергоспоживання. Можливості застосування Bluetooth на практиці безмежні: крім синхронізації з настільним комп'ютером периферії на зразок клавіатур або мишей інтерфейс дозволяє дуже просто і з невеликими витратами організувати домашню мережу. Причому вузлами цієї мережі можуть бути будь-які пристрої, що мають потребу в інформації або володіють необхідною інформацією.

2.3. Огляд окремих компонентів апаратної частини платформи

Окрім плати мікроконтролера Arduino Uno до апаратної частини будуть входити інші компоненти.

2.3.1. Bluetooth модуль для Arduino HC-05

Одне з кращих рішень для організації двостороннього зв'язку по Bluetooth вашого Arduino-пристрої з планшетом, ноутбуком або іншим Bluetooth-пристроєм - Bluetooth-модуль HC-05 (рис. 2.4), який може працювати як master (здійснювати пошук Bluetooth-пристроїв і ініціювати установку зв'язку), так і slave (пристрій - відомий).

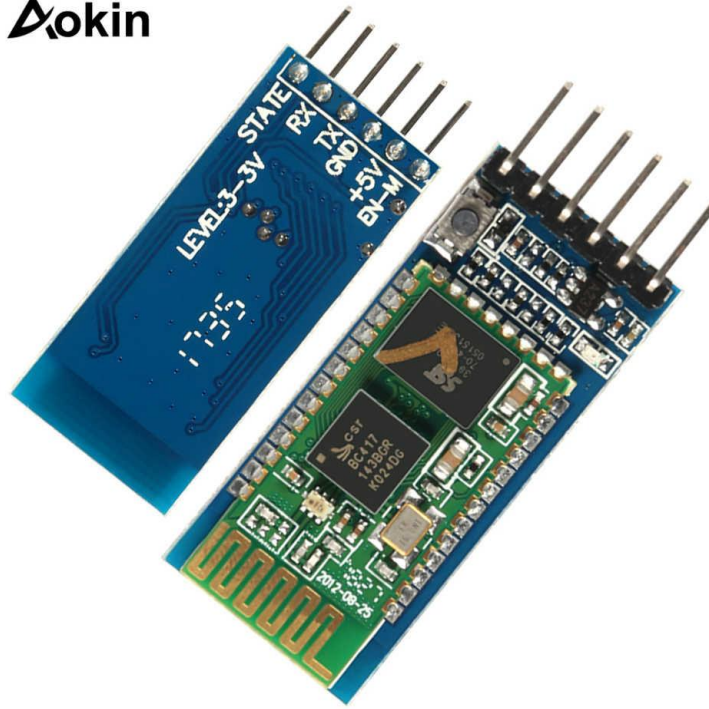


Рис. 2.4. Bluetooth-модуль HC-05.

Основні характеристики модуля HC-05:

- Діапазон частот радіозв'язку: 2,4-2,48 ГГц
- Потужність передачі: 0,25-2,5 мВт
- Чутливість: -80 dBm
- Напруга живлення: 3,3-5 В
- Струм: 50 мА
- Радіус дії: до 10 метрів
- Інтерфейс: послідовний порт

Контакти: VCC - (живлення 3,6 - 6 В); GND - (земля); TXD (надсилання даних), RXD (отримання даних) - UART інтерфейс; STATE - індикатор стану; KEY (або EN) - контакт для входу в режим програмування.

2.3.2. Драйвер мотора

Найпростіший електродвигун працює тільки на постійному струмі (від батарейки). Струм проходить по рамці, розташованій між полюсами постійного магніту. Взаємодія магнітних полів рамки з струмом і магніту змушує рамку повертатися. Після кожного півоберту колектор перемикає контакти рамки, які підходять до батарейці, і тому рамка обертається.

До мікроконтроллера мотори напряму краще не під'єднувати, адже, в основному, мотори працюють зі струмами, напруга яких більша 5В, і в момент виключення або включення мотори створюють пікові перепади струму, що може негативно вплинути на плату. Тому для кращого управління моторами використовується такий модуль як драйвер мотора. Драйвер мотора – пристрій, який дозволяє легко та зручно керувати швидкістю та напрямом обертання мотора за допомогою цифрових та ШІМ сигналів (рис. 2.5).



Рис. 2.5. Драйвер мотора.

Мотор керується трьома портами – двома цифровими та одним ШІМ. Мотор може обертатися за годинниковою стрілкою або проти неї. За це відповідають цифрові порти, які на драйвері мотора підключаються до контактів IN (рис.2.7). За

швидкість відповідає ШІМ порт і він підключається до EN контакту і може задавати діапазон значень 0-255.

IN1 - **високий**, IN2 - **низький** = обертання в одну сторону;
IN1 - **низький**, IN2 - **високий** = обертання в іншу сторону;
IN1 - **низький**, IN2 - **низький** = обертання немає;
IN1 - **високий**, IN2 - **високий** = обертання немає.

Рис. 2.6. Комбінації сигналів для мотора.

2.3.3. Сервопривід

Сервомотори Ардуіно по суті своїй відмінні пристрої, які можуть повертатися в зазначене положення і можуть застосовуватися у величезній кількості областей. Особливо зараз їх найчастіше застосовують в робототехніці. Зазвичай у них є вихідний вал, який може повертатися на 180 градусів. Використовуючи Arduino ми можемо поставити сервомоторами певне положення в яке він перейде (рис. 2.7). Сервомотор отримує сигнал, подібний PWM. Кожен цикл в сигналі триває 20 мілісекунд, і велика частина часу в значенні LOW. На початку кожного циклу значення сигналу стає HIGH на час від 1 до 2 мілісекунд. При 1 мілісекунді вона становить 0 градусів, а при 2 мілісекундах - 180 градусів, а в проміжку значення від 0 до 180. Це дуже хороший і надійний метод.

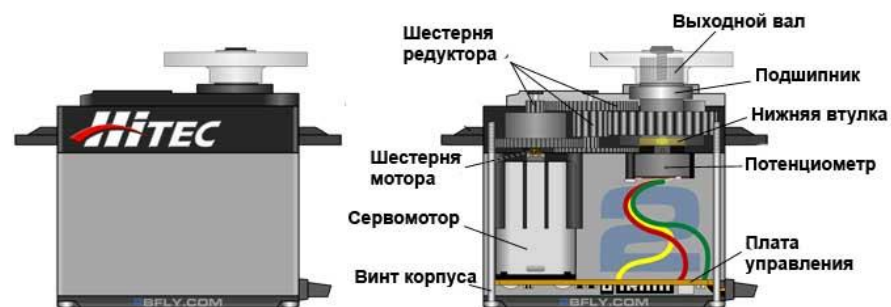


Рис. 2.7. Будова сервоприводу.

2.4. MIT App Inventor 2

MIT App Inventor - це інтегроване середовище розробки веб-додатків

Він пропонує веб-редактор "Що ви бачите - те і отримуєте" для створення програм для мобільних телефонів, орієнтованих на операційні системи Android та iOS. Він використовує блочну мову програмування, побудовану на Google Blockly і натхненню такими мовами, як StarLogo TNG та Scratch, надаючи можливість кожному створити додаток для мобільних телефонів, щоб задовольнити потреби.

Інтерфейс користувача MIT App Inventor включає два основних редактори: редактор дизайну та редактор блоків. Редактор дизайну або дизайнер (рис. 2.8) – це інтерфейс перетягування для викладення елементів користувацького інтерфейсу програми (UI). Винахідники додатків перетягують компоненти з палітри (вкрай ліворуч) до засобу перегляду (ліворуч по центру), щоб додати їх до програми. Винахідники можуть змінювати властивості компонентів (крайній праворуч). Також відображається огляд компонентів екрану та носіїв проекту (в центрі праворуч). Редактор блоків – це середовище, в якому винахідники програм можуть візуально викласти логіку своїх додатків, використовуючи кольорові блоки, які з'єднуються, як шматочки головоломки, для опису програми. Код блоків зазвичай читається зліва направо, зверху вниз.

Для сприяння розробці та тестуванню App Inventor пропонує мобільний додаток, який називається App Inventor Companion (або просто “Супутник”), який розробники можуть використовувати для тестування та регулювання поведінки своїх додатків у режимі реального часу. Таким чином, кожен може швидко створити мобільний додаток і негайно почати ітерацію та тестування.

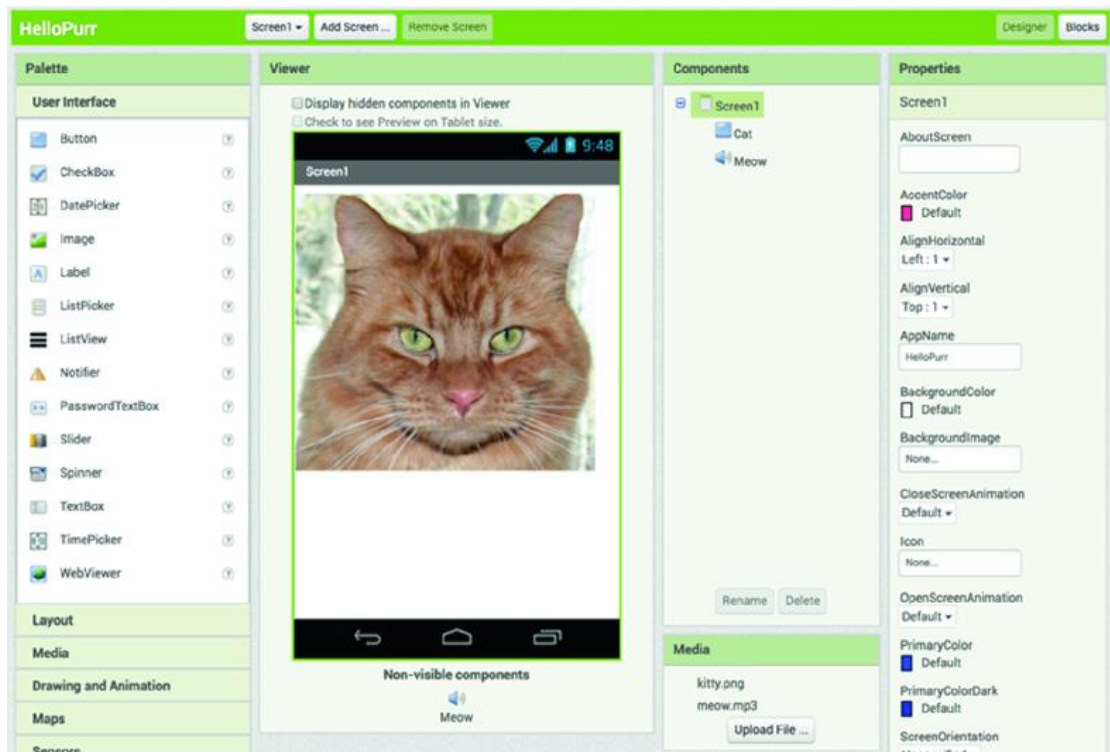


Рис. 2.8. Панель “Дизайнер” програми MIT App Inventor 2.

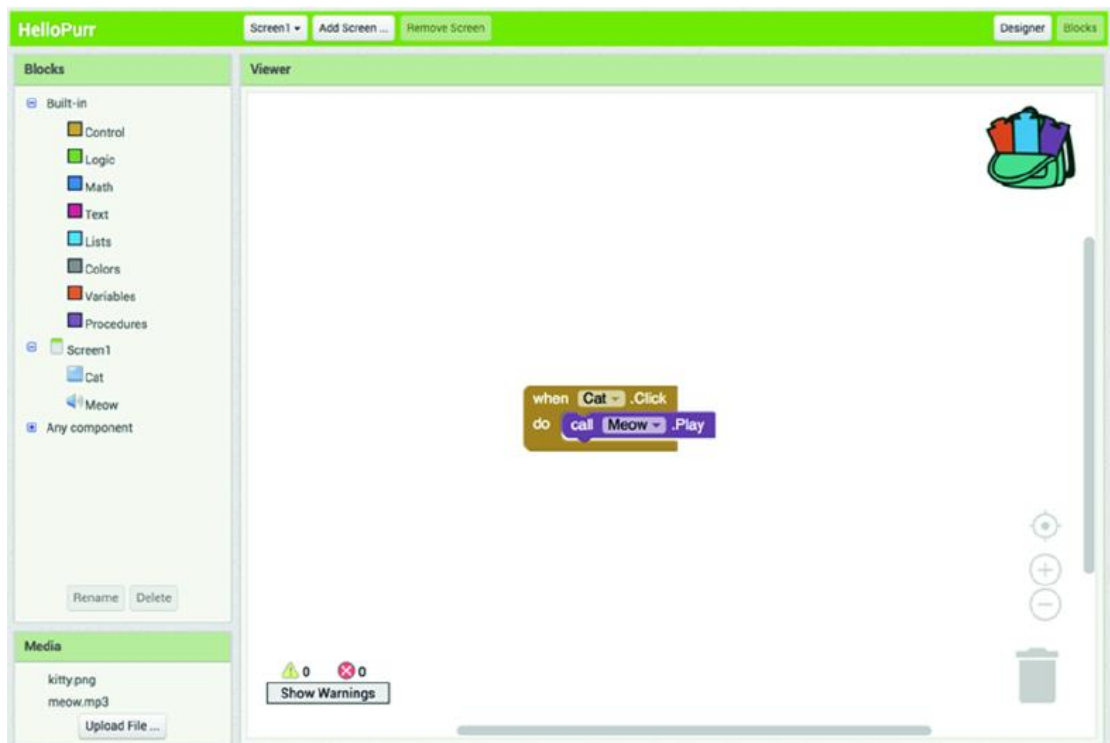


Рис. 2.9. Панель “Редактор блоків” програми MIT App Inventor 2.

2.5. Висновки до розділу 2

Для побудови моделі автомобіля з можливістю його управління необхідно реалізувати програмну та апаратну частину.

Апаратна частина – це фізична модель нашого об'єкта, а саме модель автомобіля, яка складається з безлічі деталей. Основна деталь, яка буде відповідати за управління фізичної моделі автомобіля є плата мікроконтролера Arduino UNO R3. Інші деталі будуть підключатися до портів плати, через яку будуть отримувати команди.

Програмна частина має 2 пункти, а саме розробка програмного застосунку на телефон за допомогою середовища розробки веб додатків MIT App Inventor 2, а також програмування мікроконтролера на виконання команд.

РОЗДІЛ 3

РОЗРОБКА ПРОГРАМНО-АПАРАТНИХ ЗАСОБІВ КЕРУВАННЯ МОДЕЛЛЮ АВТОМОБІЛЯ

3.1. Розробка апаратних засобів

За умовою задачі, потрібно було створити модель автомобіля з можливістю його управління.

Модель складається з наступних компонентів (рис. 3.1):

1. Мікроконтролера Arduino UNO R3;
2. Модуль драйвера мотора L298N;
3. 2-х електромоторів (3V-12V);
4. Сервопривід;
5. Блока живлення (9V-12V);
6. Перемикач
7. Bluetooth модуль HC-05;
8. Макетна плата.

Кафедра КІТ (47)				НАУ 21 38 75 000 ПЗ			
Виконав	П'ятківський В.Ю.			Розробка програмно- апаратних засобів керування моделлю автомобіля	Літера	Аркуш	Аркушів
Керівник	Климова А.С.					40	8
Консульт.					412 122		
Н-котрол.	Шевченко О.П.						
Зав. каф.	Савченко А.С.						

За допомогою електромоторів буде здійснюватись рух моделі вперед-назад. Bluetooth модуль необхідний для прийому даних від пульта управління. Драйвер мотора необхідний для управління напрямком руху моторів та швидкістю їх обертання. Сервопривід для того, щоб платформа здійснювала повороти вліво-вправо. За допомогою мікроконтролера буде відбуватися керування всіма компонентами. Макетна плата необхідна для кращого підключення пристроїв. На неї буде виводитись струм від головного джерела живлення 12V та струм від плати мікроконтролера напругою 5V. Перемикач необхідний для ввімкнення живлення.

Схематична модель підключення компонентів зображена на рис. 3.1., а схема підключень в табл. 3.1.

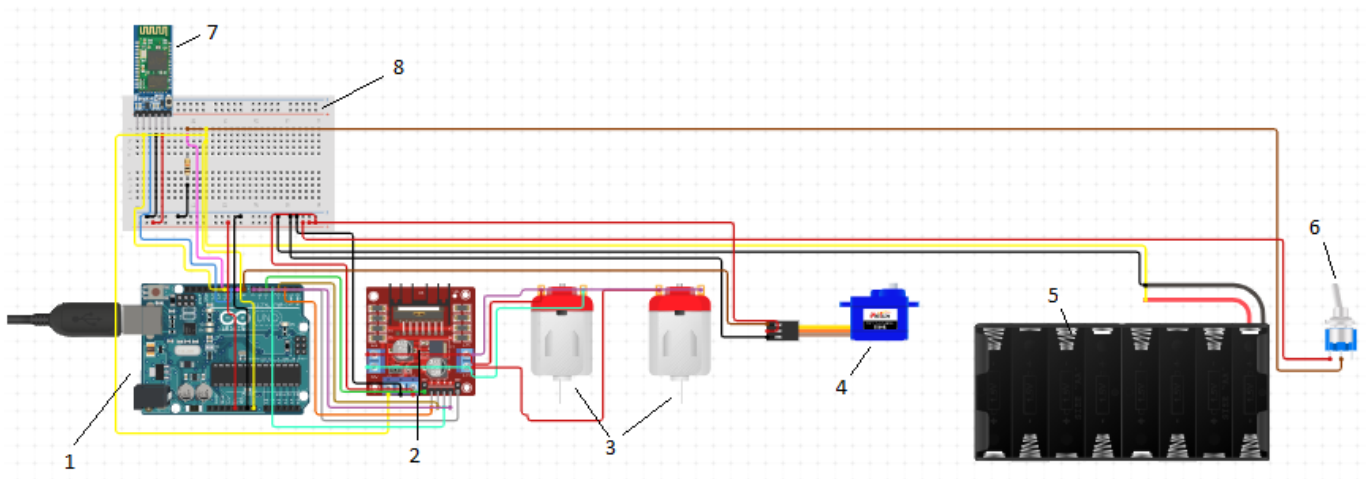


Рис. 3.1. Схематична модель підключення компонентів прототипу автомобіля на дистанційному управлінні.

Таблиця 3.1.

Таблиця підключень компонентів.

Компонент	Контакт компонента	Контакт для підключення
Драйвер мотора	Порти підключення першого мотора	Контакти електромотора 1
	Порти підключення другого мотора	Контакти електромотора 2
	Живлення для двигунів	Джерело живлення 12V
	VCC	5V мікроконтролера
	GND	GND мікроконтролера
	ENA	5 порт мікроконтролера
	INA1	2 порт мікроконтролера
	INA2	3 порт мікроконтролера
	INB1	4 порт мікроконтролера
	INB2	7 порт мікроконтролера
	ENB	6 порт мікроконтролера
Сервопривід	GND (коричневий)	GND
	VCC (червоний)	5V мікроконтролера
	Signal (жовтий)	8 порт мікроконтролера
Bluetooth модуль	GND	GND
	VCC	5V мікроконтролера
	TX	10 порт мікроконтролера
	RX	11 порт мікроконтролера

3.2. Розробка програмних засобів

Розробка програмних засобів відбувається у два етапи.

Перший етап – створення програмного інтерфейсу управління для взаємодії між телефоном та моделлю на основі мікроконтролера. Оскільки ми хочемо керувати моделлю автомобіля, який буде здійснювати рух в двох площинах, тобто вверх-вниз та вліво-вправо, тому доцільно зробити інтерфейс, з якого буде управлятися платформа, подібним до джойстика.

Для розробки графічного інтерфейсу використали середовище розробки веб додатків MIT App Inventor 2.

Найперше розташовуємо кнопки для з'єднання з пристроями Bluetooth, відключення з'єднання та показу статусу з'єднання.

Другим кроком створюємо джойстик з двох спрайтів накладених один на одного, а також надписи для виводу координат положення кнопки джойстика, за допомогою якої буде відбуватись управління рухом. Всі елементи, які ми використали, підписуємо для інтуїтивної зручності. На цьому дизайн готовий (рис. 3.2).

Далі за допомогою режиму блоків ми програмуємо поведінку нашого інтерфейсу. Для початку задамо змінні в яких будуть зберігатися початкові і поточні координати кнопки джойстика. Запрограмуємо відключення при нажатті на кнопку відключення або при збої Bluetooth. Програмуємо вибір Bluetooth та зміну надпису зі статусом. Останній крок – програмуємо поведінку джойстику, щоб він надсилав через Bluetooth координати свого положення. На цьому програмування поведінки програми закінчено (рис. 3.3).



Рис. 3.2. Інтерфейс програми для дистанційного управління.

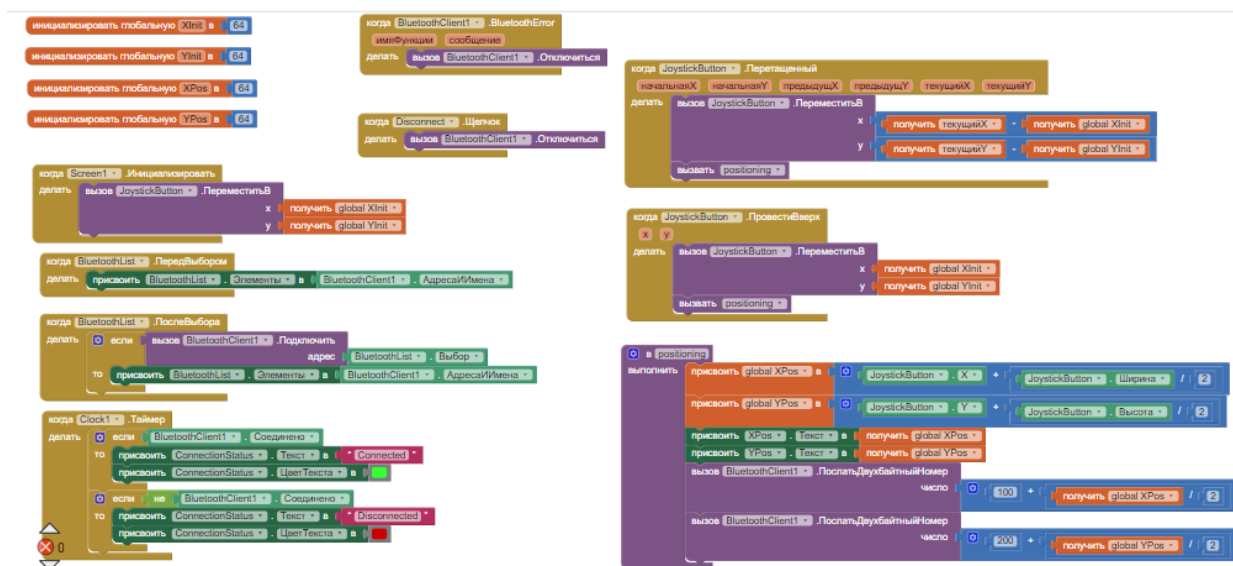


Рис. 3.3. Блоки програми для дистанційного управління.

Другий етап – програмування мікроконтролера, який буде керувати всією схемою.

Для початку розробимо загальний алгоритм, за яким буде працювати платформа (рис. 3.4).

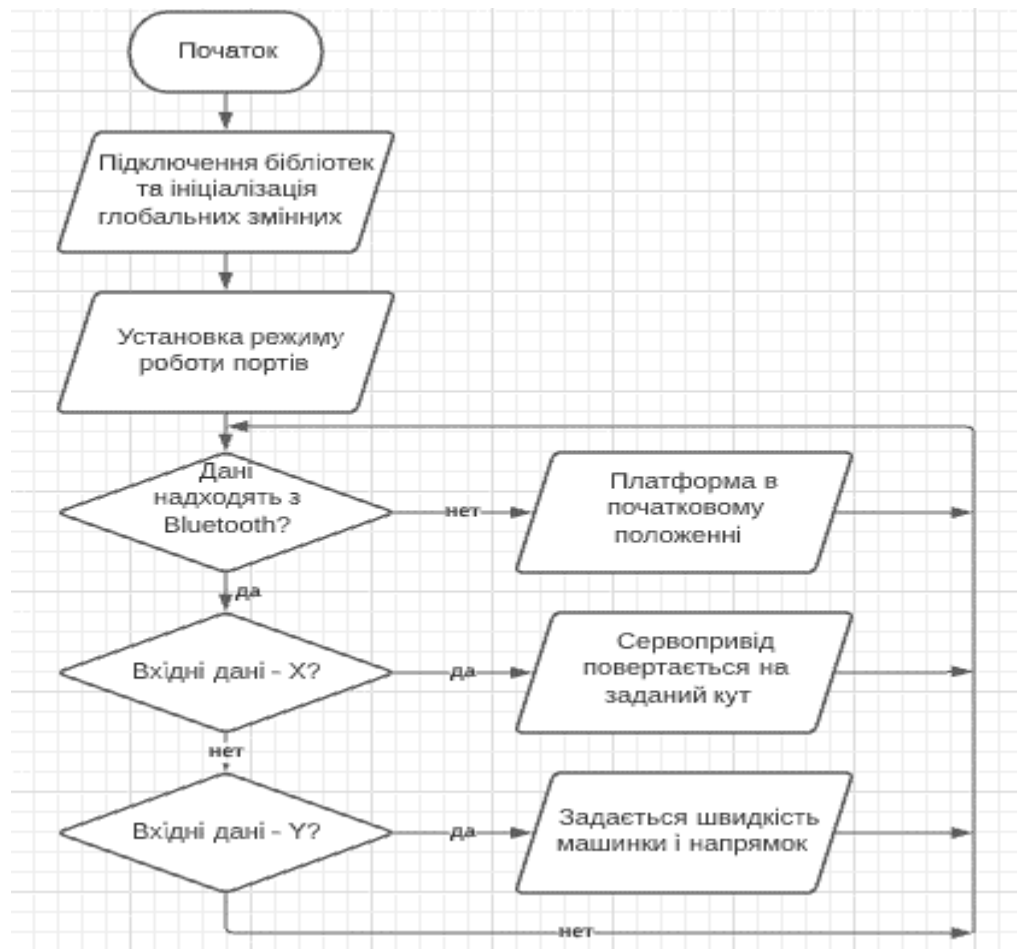


Рис. 3.4. Алгоритм роботи платформи.

На наступному кроці за допомогою середовища програмування Arduino IDE пишемо скетч для роботи мікроконтролера. Коли код програми буде готовий, зберігаємо (“Файл” -> “Зберегти”, або натиснути на зелений значок “Зберегти” зі стрілкою вниз в лівому верхньому куті інтерфейсу програми). Для того, щоб завантажити програму на плату, потрібно вгорі на панелі інструментів обрати пункт “Інструменти” та задати значення для двох підпунктів:

1. обрати плату, на яку потрібно завантажувати код програми, в нашому випадку це Arduino/Genuino UNO;

2. обрати послідовний порт, через який буде завантажуватись код програми (плата з мікроконтролером повинна бути підключеною до комп'ютера, через який буде завантажуватись код).

Останній кроком буде завантаження програми. Для цього в лівому верхньому куті потрібно натиснути на кнопку “Завантажити”. В разі успішного завантаження програма виведе повідомлення про успішне завершення завантаження в консольне вікно та дані про об'єм пам'яті мікроконтролера, яку завантажений скетч використовує (рис. 3.5).

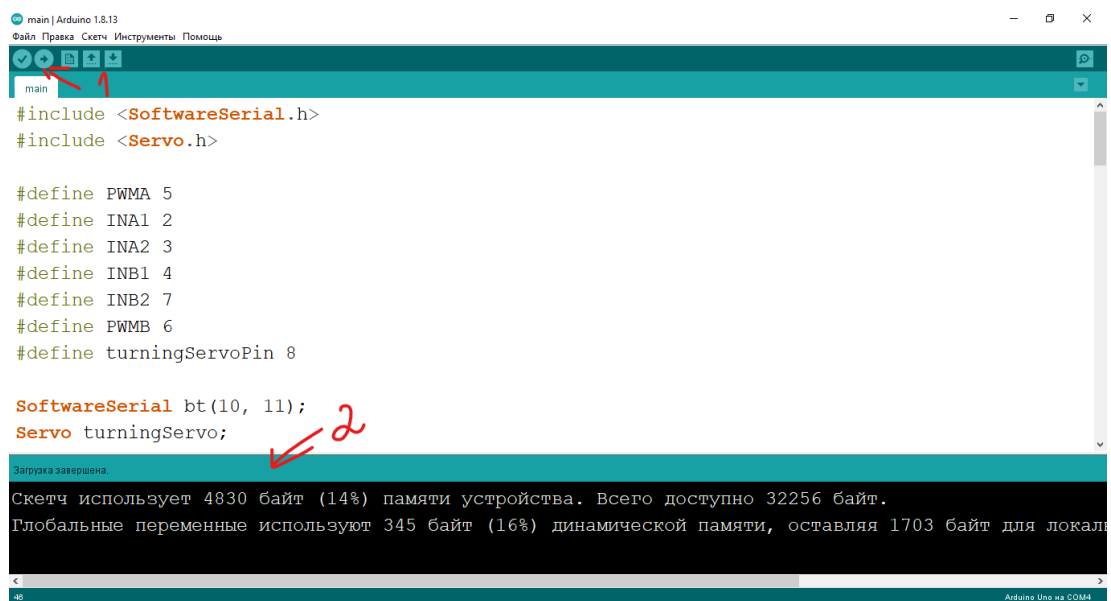


Рис.3.5. Алгоритм завантаження скетчу в плату.

3.3. Тестування засобів керування моделлю автомобіля.

Тестування проводилось за наступним алгоритмом:

1. завантаження програмного коду в мікроконтролер;
2. ввімкнення живлення моделі;
3. запуск програми на телефоні;
4. при переході в меню вибору Bluetooth відображався список доступних пристроїв Bluetooth;

5. підключення до модуля Bluetooth (зазвичай він має стандартну назву HC-05);

6. управління платформою.

Платформа приймає сигнали від телефону, та рухається правильно та без затримок.

При перевірці дальності встановлено, що платформа може приймати сигнали в радіусі 30 метрів.

Висновки до розділу 3

В розділі 3 була здійснена реалізація апаратно-програмних засобів керування моделлю автомобіля.

На першому етапі було створено схематичну модель та зібрано фізичну модель схеми.

На другому етапі було створено програмну частину моделі, а саме програму управління з телефону, та програмну прошивку для плати мікроконтролера.

На останньому етапі було протестовано систему на правильність роботи та дальність дії дистанційного управління.

ВИСНОВКИ

Всі технології, які коли-небудь створювались або будуть створюватись, необхідні для полегшення життя людей. ЕОМ є однією з них. Проте для невеликих комп'ютеризованих платформ нам необхідні дуже потужні комп'ютери, які, не маючи більші, ніж необхідні, габарити, а й витрачають забагато енергії. Тому для цього вигадали мікроконтролери, які спричинили революцію в світі електроніки, і на даний час є наявні ледь не у кожному “розумному” пристрої. І для того, щоб якомога більше спростити роботу з такими мікроконтролерами, зменшити поріг вступу в сферу схемотехніки та робототехніки, багато компаній почали випускати свої плати на мікроконтролерах, програмовані на мовах високого рівня. Однією з таких марок є Arduino, сімейство плат яких є чисельним, з платами для різноманітних варіантів використання.

Значне вкладення в розвиток людського суспільства внесло винайдення способів дистанційного управління. Завдяки цій технології проводяться багато робіт, у тому числі дослідження та спостереження критично важливі для людства.

В ході виконання даного дипломного проекту було розглянуто методи та технології побудови програмно-апаратних засобів керування моделлю автомобіля та реалізовано їх прототип.

В процесі виконання роботи було реалізовано наступні задачі:

- ознайомлення з принципами роботи дистанційного управління;
- дослідження мікроконтролера, як основного засобу реалізації моделі автомобіля;
- дослідження засобів програмування мікроконтролера;
- дослідження компонентів програмно-апаратної моделі автомобіля на дистанційному управлінні;
- реалізація прототипу платформи на дистанційному управлінні;

– побудова графічного інтерфейсу для управління моделлю автомобіля.

В даному дипломному проекті було реалізовано модель автомобіля на дистанційному управлінні на основі плати мікроконтролера, яке здійснюється за допомогою мобільного пристрою через технологію Bluetooth. Для зручнішого управління з мобільного пристрою було реалізовано програмний інтерфейс на основі середовища розробки веб додатків MIT App Inventor 2.

На завершальному етапі було протестовано прототип системи для дослідження її працездатності та виявлення помилок. Тестування проведено успішно, пристрої з'єднувались між собою, дані передавались з мобільного пристрою, модель автомобіля виконувала рух.

І, насамкінець, платформа має багато можливостей для удосконалення. Наприклад до платформи можна буде приєднати камеру для передавання зображення, або додати датчик температури чи вологості для збору атмосферних даних.

СПИСОК БІБЛЮГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Getting Started with Microcontrollers [Electronic resource]. – Режим доступу: <https://www.engineersgarage.com/> (дата звернення 20.05.2021 р.). – Назва з екрана.
2. Basics of Microcontrollers – History, Structure and Applications [Electronic resource]. – Режим доступу: <https://www.electronicshub.org/> (дата звернення 20.05.2021). – Назва з екрана.
3. Microcontroller (MCU) [Electronic resource]. – Режим доступу: <https://internetofthingsagenda.techtarget.com/> (дата звернення 20.05.2021). – Назва з екрана.
4. Виды и устройство микроконтроллеров AVR [Electronic resource]. – Режим доступу: <http://electrik.info/> (дата звернення 20.05.2021). – Назва з екрана.
5. Что такое микроконтроллеры - назначение, устройство, софт [Electronic resource]. – Режим доступу: <http://electrik.info/> (дата звернення 20.05.2021). – Назва з екрана.
6. Разработка и исследование архитектуры системы дистанционного управления бытовыми приборами [Electronic resource]. – Режим доступу: <http://masters.donntu.org/> (дата звернення 21.05.2021). – Назва з екрана.
7. Устройство дистанционного управления [Electronic resource]. – Режим доступу: <https://ru.essays.club/> (дата звернення 21.05.2021). – Назва з екрана.
8. Устройство дистанционного управления реферат по информатике , Сочинения из Информатика [Electronic resource]. – Режим доступу: <https://www.docsity.com/> (дата звернення 21.05.2021). – Назва з екрана.
9. What is the history of the remote control? [Electronic resource]. – Режим доступу: <https://science.howstuffworks.com/> (дата звернення 21.05.2021). – Назва з екрана.
10. The Most common types of Remote Control [Electronic resource]. – Режим доступу: <https://www.lripl.com/blogdetails/> (дата звернення 22.05.2021). – Назва з екрана.

11. Microcontroller Programming [Electronic resource]. – Режим доступа: <https://www.sciencedirect.com/> (дата звернення 22.05.2021). – Назва з екрана.
12. Microcontroller Programming [Electronic resource]. – Режим доступа: <https://www.build-electronic-circuits.com/> (дата звернення 22.05.2021). – Назва з екрана.
13. Arduino/Genuino UNO [Electronic resource]. – Режим доступа: <https://www.arduino.cc/> (дата звернення 23.05.2021). – Назва з екрана.
14. Плата Arduino Uno – описание, схема, распиновка [Electronic resource]. – Режим доступа: <https://voltiq.ru/> (дата звернення 23.05.2021). – Назва з екрана.
15. Плата Arduino Uno R3: схема, описание, подключение устройств [Electronic resource]. – Режим доступа: <https://arduinomaster.ru/> (дата звернення 23.05.2021). – Назва з екрана.
16. Как это работает? | Bluetooth [Electronic resource]. – Режим доступа: <https://hi-news.ru/> (дата звернення 24.05.2021). – Назва з екрана.
17. Bluetooth: технология и ее применение [Electronic resource]. – Режим доступа: <https://www.ixbt.com/> (дата звернення 24.05.2021). – Назва з екрана.
18. Что такое Bluetooth и как он работает? [Electronic resource]. – Режим доступа: <http://1234g.ru/> (дата звернення 24.05.2021). – Назва з екрана.
19. Устройство и принцип работы простейшего электродвигателя [Electronic resource]. – Режим доступа: <https://voltiq.ru/> (дата звернення 25.05.2021). – Назва з екрана.
20. Драйверы двигателя L298N, L293D и Arduino Motor Shield [Electronic resource]. – Режим доступа: <https://arduinomaster.ru/> (дата звернення 25.05.2021). – Назва з екрана.
21. Arduino и модули Bluetooth HC-05/06 [Electronic resource]. – Режим доступа: <https://voltiq.ru/> (дата звернення 26.05.2021). – Назва з екрана.
22. MIT App Inventor: Objectives, Design, and Development [Electronic resource]. – Режим доступа: <https://link.springer.com/> (дата звернення 26.05.2021). – Назва з екрана.

ДОДАТКИ

Додаток А

Скетч для керуючого мікроконтролера

```
#include <SoftwareSerial.h>
```

```
#include <Servo.h>
```

```
#define PWMA 5
```

```
#define INA1 2
```

```
#define INA2 3
```

```
#define INB1 4
```

```
#define INB2 7
```

```
#define PWMB 6
```

```
#define turningServoPin 8
```

```
SoftwareSerial bt(10, 11);
```

```
Servo turningServo;
```

```
int xAxis = 64;
```

```
int yAxis = 64;
```

```
bool forward;
```

```
void setup()
```

```
{
```

```
    pinMode(PWMA, OUTPUT);
```

```
    pinMode(INA1, OUTPUT);
```

```
    pinMode(INA2, OUTPUT);
```

```
    pinMode(INB1, OUTPUT);
```

```
pinMode(INB2, OUTPUT);
pinMode(PWMB, OUTPUT);
turningServo.attach(turningServoPin);
bt.begin(9600);
Serial.begin(9600);
}

void loop()
{
  if (bt.available())
  {
    int input = bt.read();
    if (input > 100 && input < 200)
    {
      input = input - 100;
      xAxis = map(input, 32, 96, 0, 180);
      turningServo.write(xAxis);
    }
    else if (input > 200 && input < 300)
    {
      input = input - 200;
      if (input >= 64)
      {
        forward = true;
        input = input - 64;
      }
      else if (input < 64)
      {
        forward = false;
```

```

    }
    yAxis = map(input, 0, 32, 0, 255);
    moving(forward);
    setSpeed(yAxis);
  }
  delay(100);
}
else
{
  setSpeed(0);
  turningServo.write(90);
}
}

```

```

void moving(bool direction)
{
  digitalWrite(INA1, !direction);
  digitalWrite(INA2, direction);
  digitalWrite(INB1, !direction);
  digitalWrite(INB2, direction);
}

```

```

void setSpeed(int speed)
{
  analogWrite(PWMA, speed);
  analogWrite(PWMB, speed);
}

```